### Configure SSH on the Remote Agent

1. View the home directory contents:
2. [cloud\_user@jenkins ~]$ ls

Once you see a SERVER-READY file listed, continue with the rest of the lab.

1. Log in to the agent node using the credentials provided:
2. [cloud\_user@jenkins ~]$ ssh cloud\_user@<AGENT\_PUBLIC\_IP\_ADDRESS>
3. Create the directory for the user's home:
4. [cloud\_user@agent ~]$ sudo mkdir /var/lib/jenkins
5. Add the user, assigning the home directory:
6. [cloud\_user@agent ~]$ sudo useradd -d /var/lib/jenkins jenkins

You may get an error message that the home directory already exists and it isn't copying any file from the skel directory. You can ignore this error, as it doesn't affect the lab.

1. Make the user the owner of their home directory:
2. [cloud\_user@agent ~]$ sudo chown -R jenkins:jenkins /var/lib/jenkins
3. Create an .ssh directory for the jenkins user:
4. [cloud\_user@agent ~]$ sudo mkdir /var/lib/jenkins/.ssh
5. Generate an SSH key:
6. [cloud\_user@agent ~]$ ssh-keygen
7. Press **Enter** to accept defaults until it completes.
8. View the contents of ~/.ssh/id\_rsa.pub:
9. [cloud\_user@agent ~]$ cat ~/.ssh/id\_rsa.pub
10. Copy the output, and paste it into a text file.
11. Create a /var/lib/jenkins/.ssh/authorized\_keys file:
12. [cloud\_user@agent ~]$ sudo vim /var/lib/jenkins/.ssh/authorized\_keys
13. Press i to enter insert mode.
14. Paste in the id\_rsa.pub contents you just copied.
15. Save and exit the file by pressing **Escape** followed by :x.
16. View the contents of the id\_rsa private key:
17. [cloud\_user@agent ~]$ cat ~/.ssh/id\_rsa
18. Copy the entirety of the output (including the BEGIN RSA PRIVATE KEY and END RSA PRIVATE KEY lines), and paste it into a text file.
19. Exit the agent node:
20. [cloud\_user@agent ~]$ exit
21. Create an .ssh directory on the controller in the jenkins directory:
22. [cloud\_user@jenkins ~]$ sudo mkdir /var/lib/jenkins/.ssh
23. Copy the known\_hosts entry over from the .ssh directory in /home/cloud\_user to the jenkins user's .ssh directory:

[cloud\_user@jenkins ~]$ sudo cp ~/.ssh/known\_hosts /var/lib/jenkins/.ssh

**Set Up the Node on the Jenkins Controller**

1. Navigate to the Jenkins controller web interface: http://<PUBLIC\_IP\_OF\_JENKINS\_CONTROLLER>:8080
2. In the terminal, get the initial admin password:
3. [cloud\_user@jenkins ~]$ sudo cat /var/lib/jenkins/secrets/initialAdminPassword
4. Copy the output, and paste it into the **Administrator password** field in the Jenkins web interface.
5. Click **Continue**.
6. Click **Install suggested plugins**.
7. On the **Create First Admin User** screen,
   * **Username:** Enter whatever you'd like (e.g., *student*).
   * **Password:** Use the same one as the provided Jenkins controller password on the lab page
   * **Full name:** Enter whatever you'd like.
   * **E-mail address:** Enter a fake email address.
8. Click **Save and Continue**.
9. Click **Save and Finish**.
10. Click **Start using Jenkins**.
11. On the **Manage Jenkins** page, click **Manage Nodes and Clouds**.
12. In the left-hand menu, click **New Node**, and set the following values:
    * **Node name:** Enter *worker1*.
    * **Permanent Agent:** Select.
    * Click **Create**.
    * **#of executors:** Enter *2*.
    * **Remote root directory:** Enter */var/lib/jenkins*.
    * **Labels:** Enter *linux*.
    * **Launch method:** Select **Launch agents via SSH**.
    * **Host:** Enter the public IP address of the agent node.
    * **Credentials:** Click **Add** > **Jenkins**.
      + **Kind:** Select **SSH Username with private key**.
      + **Description:** Enter *jssh*.
      + **Username:** Enter *jenkins*.
      + **Private Key:** Select **Enter directly**, click **Add**, and paste in the id\_rsa private key you copied earlier.
      + Click **Add**.
    * **Credentials**: Click **- none -**, and select **jenkins (jssh)**.
13. Leave the rest as their defaults, and click **Save**.
14. On the **Nodes** page, click **worker1**.
15. In the left-hand menu, click **Log** to make sure there was a successful SSH authentication.
    * If you see it in the log, and you see an entry that says *Agent successfully connected and online*, then it all worked correctly.

**Test a Remote Build**

1. In the upper left corner, click **Jenkins**.
2. In the left-hand menu, click **Manage Jenkins**.
3. Click **Configure System**, and then set the following values:
   * **Labels:** Enter *jenkins*.
   * **Usage**: Select **Only build jobs with label expressions matching this node**.
4. Click **Save**.
5. Click **New Item**.
6. Under **Enter an item name**, enter *test*.
7. Select **Freestyle project**.
8. Click **OK**.
9. Under **Build Steps**, click **Add build step** > **Execute shell**.
10. In the **Command** box, enter *hostname > location.txt*.
11. Under **Post-build Actions**, click **Add post-build action** > **Archive the artifacts**.
12. In the **Files to archive** box, enter *location.txt*.
13. Click **Advanced**.
14. Check the box next to **Fingerprint all archived artifacts**.
15. Click **Save**.
16. In the left-hand menu, click **Build Now**.
17. Under **Build History**, click the build number.
18. In the left-hand menu, click **Console Output** to see the build process happening.
    * You should see that it's building remotely on worker1 on the test workspace, and it should say it's archiving artifacts and recording fingerprints.
19. In the breadcrumb navigation at the top of the screen, click **test**.
    * You should see location.txt listed under **Last Successful Artifacts**.
20. Right-click **view** to open it in a new browser tab.
    * You should see the agent hostname.

**Conclusion**